

## (12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property  
Organization  
International Bureau



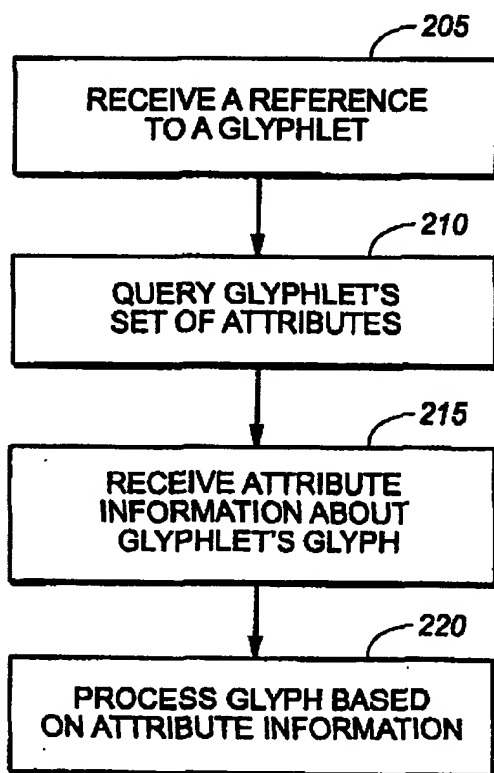
(43) International Publication Date  
5 February 2004 (05.02.2004)

PCT

(10) International Publication Number  
**WO 2004/012099 A2**

- (51) International Patent Classification<sup>7</sup>: G06F 17/21 (72) Inventor: **RENNER, John, S.**; 2 Lane Place, Atherton, CA 94027-3010 (US).
- (21) International Application Number: PCT/US2003/024111 (74) Agent: **MOLLBORN, Fredrik**; Fish & Richardson P.C., 500 Arguello Street #500, Redwood City, CA 94063 (US).
- (22) International Filing Date: 31 July 2003 (31.07.2003) (81) Designated State (national): JP.
- (25) Filing Language: English (84) Designated States (regional): European patent (AT, BB, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR).
- (26) Publication Language: English
- (30) Priority Data: 10210,512 31 July 2002 (31.07.2002) US
- (71) Applicant: **ADOBE SYSTEMS INCORPORATED** [US/US]; 345 Park Avenue, San Jose, CA 95110 (US).
- Published:  
— without international search report and to be republished upon receipt of that report
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: GLYPHLETS



(57) Abstract: Methods and apparatus, including computer program products, implementing and using techniques for processing and constructing an electronic text document. An electronic document is provided that includes a string (100) having one or more references. The string is parsed to identify a reference (110). A glyphlet is identified based on the identified reference (110). The glyphlet includes a set of character attributes defining semantic information of a character (120) and a set of glyph attributes defining appearance information for a representation of the character (120). One or more character attributes or glyph attributes for the identified glyphlet are used to process text in the electronic document.

Best Available Copy

## GLYPHLETS

## BACKGROUND OF INVENTION

This invention relates to providing glyphs in a text document.

A character is the smallest component of written language having semantic value.

5 A character refers to an abstract meaning and/or shape, rather than a specific shape. A glyph is a representation of a character. A glyph image is the actual concrete image of a glyph representation having been rasterized or otherwise imaged onto some display surface.

An encoded character is a character that is associated with an encoding value, for  
10 example, a scalar value included in a character set standard such as ASCII (American Standard Code for Information Interchange) or Unicode. An encoding value maps to a set of character attributes defining semantic information of the character. Character set standards are defined by standards organizations: for example, the ASCII standard is defined by ANSI, and the ISO Standard 8859 is defined by ISO (International Standards  
15 Organization). Character set standards are generally revised from time to time. Typically, when a character set standard is defined, the encoding values are simultaneously defined.

Character attributes can include one or more of the following: character case, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic  
20 character. Other character attributes are possible.

A glyph can be associated with a set of glyph attributes defining appearance information for a representation of the corresponding character. Glyph attributes can include one or more of the following: glyph shape, glyph metrics, typeface name, glyph baseline and glyph kerning. Generally, glyph attributes provide the information necessary  
25 to render the glyph image.

A font is a collection of glyphs and a corresponding encoding mapping. A font is typically constructed to support a character set standard. That is, fonts include glyphs representing characters included in the character set standard. When the character set standard is revised, the font manufacturer may need to revise the font to accommodate the  
30 changes, including the addition of new glyphs. In that case, a new font is re-issued conforming to the new character set standard. Revising fonts is costly for the designer and inconvenient for users who must track versions of the font and determine whether or not they have fonts supporting the latest character set standard.

Text documents typically include a text string that includes one or more encoding values that represent characters in the text. An encoding value can map to a character in a character set standard and to a glyph in a font constructed to support the character set standard. Thus, a text engine (e.g., a word processing application) processing an electronic document that includes a text string of encoding values can obtain character attribute information about an encoded character represented by the encoding value by mapping the encoding value to the character set standard. The text engine can also render a representation of the character, that is, a glyph image, based on glyph attribute information obtained from a specified font, using the same encoding value. The encoding value-attribute associations are typically available for a text engine to reference by looking them up in fixed and static tables, indexed by encoding value. The attributes are not part of the document itself, but are usually built into the text engine or the operating system used by the application.

A character can be processed based on its character attributes and/or glyph attributes. For example, a layout engine that is setting text in vertical writing mode might handle numerals in a specialized way, or might handle a currency symbol differently than numerals in some contexts. As another example, attributes can be critical for input methods, as the user may need to choose the character based on the corresponding glyph's radical, stroke-count or pronunciation (e.g., a software agent used to assist selecting Chinese/Japanese characters). Thus, for a representation of a character (i.e., a glyph) to participate fully in an electronic document, the character and glyph attributes of the character and corresponding glyph must be accessible by a text engine processing the electronic document.

25

## SUMMARY

The present invention provides methods and apparatus, including computer program products, for processing and constructing an electronic text document. In general, in one aspect, the invention features providing an electronic document including a string that includes one or more references and parsing the string to identify a reference. Based on the identified reference, a glyphlet is identified including a set of character attributes defining semantic information of a character and a set of glyph attributes defining appearance information for a representation of the character. One or more

30

character attributes or glyph attributes for the identified glyphlet are used to process text in the electronic document.

Implementations of the invention can include one or more of the following. The string can include a plurality of references comprising one or more in-band values defined in an encoding standard. Parsing the string to identify a reference can include interpreting a plurality of in-band values to define the identified reference. Interpreting a plurality of in-band values to define the identified reference can include identifying one or more target attributes, and identifying a glyphlet based on the identified reference can include identifying a glyphlet in a collection of glyphlets based on the identified target attributes. The collection of glyphlets can be embedded within the electronic document or can be external to the electronic document.

Alternatively, the plurality of in-band values can define the identified glyphlet. In another alternative, the plurality of in-band values can identify a location external to the electronic document from which the identified glyphlet can be retrieved.

In another implementation, the string can include one or more references comprising one or more out-of-band values not defined in an encoding standard. The identified reference can include one or more of the out-of-band values. If the identified glyphlet is embedded within the electronic document, the one or more out-of-band values can be directly associated with the identified glyphlet. Identifying a glyphlet based on the identified reference can include identifying one or more target attributes based on the identified reference and identifying a glyphlet in a collection of glyphlets based on the identified target attributes. The collection of glyphlets can be embedded within the electronic document or external to the electronic document. The one or more out-of-band values can identify a location external to the electronic document from which the identified glyphlet can be retrieved.

The set of character attributes can include one or more character attributes selected from the group consisting of character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character. The set of glyph attributes can include one or more glyph attributes selected from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline and glyph kerning.

The identified glyphlet can be retrieved from a memory external to the electronic document, and can be retrieved from a collection of glyphlets. Alternatively, the

identified glyphlet can be retrieved from storage embedded within the electronic document, including from a collection of glyphlets stored within the electronic document.

In general, in another aspect, the invention features method and apparatus for constructing a text document. User input is received selecting a character, and a glyphlet  
5 corresponding to the selected character is identified. The glyphlet includes a set of character attributes defining semantic information of the selected character and a set of glyph attributes defining appearance information for a glyph representative of the selected character. A reference to the identified glyphlet is inserted into a text document.

Implementations of the invention can include one or more of the following. The  
10 identified glyphlet can be embedded in the text document. The user input selecting a character can include user input selecting a glyph shape representing the character. The reference to the identified glyphlet can include one or more in-band values defined in an encoding standard. The one or more in-band values can define one or more target attributes uniquely identifying the identified glyphlet in a collection of glyphlets.  
15 Alternatively, the one or more in-band values can define the identified glyphlet.

The reference to the identified glyphlet can include one or more out-of-band values not defined in an encoding standard. The one or more out-of-band values can be associated with one or more target attributes uniquely identifying the identified glyphlet in a collection of glyphlets. Alternatively, if the identified glyphlet is embedded in the  
20 text document, the one or more out-of-band values can be directly associated with the identified glyphlet.

In general, in another aspect, the invention features method and apparatus for representing a character in a text document. A reference identifying a glyphlet is inserted into the text document. The identified glyphlet includes a set of character attributes  
25 defining semantic information of a character and a set of glyph attributes defining appearance information for a representation of the character.

Implementations of the invention can include one or more of the following. The identified glyphlet can be embedded in the text document. The reference can include one or more in-band values defined in an encoding standard that are interpreted to identify  
30 one or more target attributes from which the identified glyphlet can be identified. Alternatively, the reference can include one or more in-band values that are interpreted to define the identified glyphlet.

In another implementation, the reference can include one or more out-of-band values not defined in an encoding standard. The one or more out-of-band values can

identify one or more target attributes, and the reference can identify the glyphlet from a collection of glyphlets based on the one or more target attributes. Alternatively, if the identified glyphlet is embedded in the text document, the one or more out-of-band values can be directly associated with the identified glyphlet.

5       The set of character attributes can include one or more character attributes selected from the group consisting of character case, character case, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character. The set of glyph attributes can include one or more glyph attributes selected from the group  
10       consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline, and glyph kerning.

      In general, in another aspect, the invention features a glyphlet. A glyphlet is a data structure stored on a computer readable medium. The data structure includes character data representing one or more character attributes defining semantic information  
15       of a character and glyph data representing one or more glyph attributes defining appearance information for a representation of the character.

      Implementations of the invention can include one or more of the following. The one or more character attributes can include one or more character attributes selected from the group consisting of character case, character category, character combining  
20       class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character. The one or more glyph attributes can include one or more glyph attributes selected from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline, and glyph kerning.

25       In general, in another aspect, the invention features an electronic document stored on a computer readable medium. The electronic document includes electronic data defining a string that includes one or more references identifying glyphlets. A glyphlet includes a set of character data representing one or more character attributes defining semantic information of a character, and a set of glyph data representing one or more  
30       glyph attributes defining appearance information for a representation of the character.

      Implementations of the invention can include one or more of the following. The electronic document can further include electronic data defining a collection of one or more glyphlets identified by the references in the string. A reference can include one or more in-band values defined in an encoding standard that are interpreted to identify one

or more target attributes from which a glyphlet can be identified. Alternatively, a reference can include one or more in-band values that are interpreted to define a glyphlet.

In another implementation, a reference can include one or more out-of-band values not defined in an encoding standard. The one or more out-of-band values can  
5 identify one or more target attributes, and the reference can identify a glyphlet from a collection of glyphlets based on the one or more target attributes. The electronic data can define a collection of one or more glyphlets identified by the references in the string, and one or more out-of-band values can be directly associated with one or more glyphlets in the collection.

10 The set of character attributes can include one or more character attributes selected from the group consisting of character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character. The set of glyph attributes can include one or more glyph attributes selected  
15 from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline, and glyph kerning.

The invention can be implemented to realize one or more of the following advantages. Because a glyphlet (including glyph attributes and character attributes) can be identified based on a reference, a text engine can access glyph and character attribute  
20 information about the glyphlet without reliance on a specific encoding mapping. The text engine can process the glyphlet fully as any glyph included in an encoding mapping to a character set standard. Identifying a glyphlet based on a reference to a set of attributes adds a level of searchable access to glyphs beyond the traditional one-to-one encoding mapping. A target glyph can be stored external to a font. A font can be expandable by  
25 having access to additional glyph shapes when used in conjunction with a collection of one or more glyphlets, which glyphlets are accessible by a text engine processing a document that includes text in the font. The ability to use a collection of glyphlets in conjunction with a font can eliminate the need to create, distribute and install a revised font including additional glyphs, which is cost-effective, convenient and efficient to both  
30 font manufacturers and users.

The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features and advantages of the invention will be apparent from the description, the drawings, and the claims.

## DESCRIPTION OF DRAWINGS

FIG 1 represents a text string including an in-band value reference and multiple encoded characters.

FIG 2 is a flowchart showing a process for processing text that includes a reference to a glyphlet.

FIG 3 is a flowchart showing a process for identifying a glyphlet based on a reference to a set of attributes.

FIG 4 is a flowchart showing a process for constructing a text document including a glyphlet.

FIG 5 is a flowchart showing a process for identifying a glyph referenced by a dynamic font.

Like reference symbols in the various drawings indicate like elements.

## DETAILED DESCRIPTION

A glyph that does not represent a character in a character set standard, or that does not have an encoding value in a font and is therefore not associated with a character in a character set standard, is sometimes referred to as a gaiji (a Japanese term meaning "foreign glyph"). Because the character set standard does not include a mapping from an encoding value to a set of character attributes for a gaiji, there are no character attributes associated with the gaiji. Currently, if a user wants to insert a gaiji into a document, a separate graphic image of the gaiji's glyph shape can be created and inserted into a gap in the layout of the text as displayed by a text-processing application. The display of the glyph shape gives the illusion that the gaiji is part of the text, but the text's underlying string itself does not include the gaiji or a reference to the gaiji. Further, the gaiji cannot participate in text-processing activities, for example, select, find/replace, spell-check and the like, as glyphs representing encoded characters can, because the text-processing application has no information about the character attributes associated with the gaiji. If the gaiji is not included in a font, the text-processing application may also have no information about the glyph attributes associated with the gaiji.

If character and glyph attributes are stored with a glyph, a text engine need not depend on any specific encoding to determine the character and glyph attributes associated with the glyph. Accordingly, such a glyph can participate fully as a glyph representing an encoded character, even though the glyph is not mapped to a set of glyph



attributes by the font, and is not mapped to a set of character attributes in a character set standard.

A glyphlet is a set of glyph attributes and a set of character attributes. Glyph attributes define appearance information for a representation of a character, and can include: glyph shape, typographic weight, typographic width, slant, number of strokes ,  
5 glyph metrics, typeface name, glyph baseline and glyph kerning. Character attributes define semantic information of a character, and can include: character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and  
10 ideographic character. The glyph and character attribute lists above are not exhaustive and other attributes are possible.

A glyphlet provides a direct relationship between a representation of a character (*i.e.*, a glyph) and the associated character attributes and glyph attributes. The character and glyph attributes can be accessed if the identity of the corresponding glyphlet is  
15 known. By contrast, a glyph included in a font is only indirectly related to a set of character attributes associated with the glyph. The character attributes are defined by the character set standard, as described above. The relationship between a glyph in a font and a set of character attributes in a character set standard is therefore formed through the encoding mapping of the character set standard and the font. Aside from this relationship,  
20 the glyph and set of character attributes are independent of one another. If the relationship does not exist, for example, if a font includes a glyph that does not have a corresponding mapping in the character set standard used to construct the font, then a text engine cannot access character attribute information for the glyph. Without accessible character attributes, the glyph cannot participate fully as a character within a text  
25 document.

A glyphlet can be implemented as a data structure storing character attributes and glyph attributes, or surrogates for them, *e.g.*, pointers or indices. In one implementation, a glyphlet is packaged as a "sfnt" (Open Type) structured font, including at least two tables: one for the glyph shape and one for the metadata including the character and glyph  
30 attributes. The metadata table is an indexed list of attribute key-value entries. The glyphlet can be queried for an attribute by searching the metadata list for entries whose key matches the desired attribute. The glyphlet's metadata can also be pre-fetched and cached in a database, which can be queried more efficiently than inspecting each glyphlet's sfnt structure directly.

A glyphlet included in a text document can be represented by a "reference" included in a text string: a text string can include references in addition to encoding values representing encoded characters. Processing an electronic document including a glyphlet can be implemented using a text engine configured to recognize a reference from which a glyphlet can be identified, and to interpret the reference to identify the glyphlet, as described further below.

A reference from which a glyphlet can be identified can be an out-of-band value not defined in an encoding standard. For example, a reference can be an integer value recognizable by a text engine as referring to a glyphlet that is not included in an encoding mapping. The out-of-band value can be directly associated with a glyphlet embedded within the electronic document. Alternatively, the out-of-band value can be associated with information indicating where a glyphlet can be found external to the text document, for example, an address to a server from which the glyphlet can be downloaded.

In another alternative, the out-of-band value can be associated with one or more target attributes, embedded within the electronic document, that uniquely identify a glyphlet. The target attributes can form the basis of a query used to query a collection of glyphlets, from which the glyphlet can be identified. The collection of glyphlets can be embedded within the document, or information can be embedded in the document indicating an external store where the collection of glyphlets can be found.

In any event, the reference is selected to be recognizable by an appropriately configured text engine as a reference from which a glyphlet can be identified, such that the text engine must look somewhere other than a character set standard for attribute information necessary to process the glyphlet. For example, an integer or range of integers can be set aside for use as references only, such that a text engine processing a string including encoding values included in an encoding standard and references, will recognize an integer within the range as a reference and identify a glyphlet accordingly.

In another implementation, the reference can be one or more in-band values defined in an encoding standard. The in-band values can define a glyphlet, that is, include the glyph attributes and character attributes. Alternatively, the in-band values can identify where a glyphlet can be found external to the text document, for example, an address to a server from which the glyphlet can be downloaded.

In another alternative, the in-band values can define one or more target attributes that can be used to identify the glyphlet. The target attributes defined by the in-band value can be used to form a query. Using the query, the glyphlet can be identified from a

collection of glyphlets: the glyphlet including one or more attributes satisfying the query. As discussed above, the collection of glyphlets can be embedded in the text document, or can be found in an external store identified by information embedded in the text document.

5 In-band values can include, for example, an XML string including one or more tagged elements. The tagged elements can be attribute-value pairs defining target attributes. FIG 1 shows a representation of a text string 100 including several encoded characters 105 and an in-band reference 110 from which a glyphlet can be identified. In a text document, the text string 100 would be formed of encoding values, for example,  
10 ASCII values associated with each of the encoded characters 105 and the encoded characters included in the in-band value reference 110. For illustrative purposes, the representative characters are shown in FIG 1.

In this example, the reference 110 includes attribute information from which a glyphlet can be identified. For example, the character category (charcat) attribute has the  
15 value "Currency Symbol". As described above, the attributes defined by the reference 110 can be used to form a query. A collection of glyphlets can be queried and a glyphlet having one or more attributes satisfying the query can be identified. A glyph shape representing the character having the attributes defined in the reference 110 can be rendered using glyph attribute information accessible from the identified glyphlet. In this  
20 example, the glyph shape for the Euro symbol, € 120, can be rendered and displayed by a display device, such as a monitor or printer, along with glyph shapes representing the encoded characters 105 included in the text string 100.

One or more glyphlets can be included in a cache included within a text document, and, as discussed above, a reference can be a pointer or offset to a glyphlet  
25 within the cache. Alternatively, a reference can be a pointer to a location in memory storing the glyphlet.

FIG 2 shows a process for retrieving attribute information from a glyphlet, for example, using a text engine processing a text document. As discussed above, strings within the text document include both encoding values associated with encoded  
30 characters, and references associated with glyphlets. In the first step, a reference is encountered in a string and recognized as a reference from which a glyphlet can be identified (Step 205). The recognition occurs because, for example, the reference is an out-of-band value integer within a range of integers reserved for references, or because the reference is one or more in-band values defining target attributes of a glyphlet. If the

text engine requires character or glyph attribute information for the glyphlet, then the text engine queries the glyphlet (Step 210). Attribute information is retrieved from the glyphlet in response to the query (Step 215). For example, a text engine might be tasked to build a concordance for a text document, which requires a count of words within the document. In order to identify a word formed by a string, the text engine may need to retrieve character attribute information for a glyphlet represented by a reference included in the string. For example, the text engine may need to determine the pronunciation of the glyphlet, in order to identify the word. The string including the glyphlet can then be processed based on the retrieved attribute information (Step 220).

As discussed above, a reference can be used to identify one or more target attributes, from which a target glyphlet can be identified. FIG 3 shows a process for identifying a glyphlet based on a reference to a set of attributes. For illustrative purposes, consider a text engine processing a text document. The text engine receives a reference, which can be included in a string among encoding values associated with encoded characters, and recognizes the reference as referring to a set of attributes (Step 305). As discussed above, the reference can be recognizable as referring to a set of attributes because, for example, the reference is an integer within a range of integer values reserved for references. Alternatively, the reference can specify attributes, for example, the reference can be a tagged XML string (included in the text string) specifying a set of attributes. To identify a glyphlet associated with the reference, the text engine generates a query based on the set of attributes identified by the reference (310).

A collection of one or more glyphlets is then queried, to identify a glyphlet including one or more attributes satisfying the query (315). It is possible that two or more glyphlets satisfy a query, in which case user can be presented a visual representation of the glyphlets and prompted to make a choice. Alternatively, it is possible that no glyphlet satisfies a query. In this case, the text engine can be configured to alert the user that no corresponding glyphlet is available, e.g., by displaying an error message or, if appropriate, inserting a default glyph at the relevant location in the text. The collection of glyphlets can be a cache of glyphlets included in the text document, or can be stored in memory, separate from the text document. A glyphlet including one or more attributes satisfying the query is identified (Step 320). The text engine can then access information from the glyphlet to render the glyph (i.e., glyph attributes) or otherwise process the glyphlet (i.e., character attributes).

The independence of a glyphlet from a character set standard or a font allows new characters to be introduced into a text processing application with relative ease. By way of illustrative example, consider the introduction of a common currency in Europe, which required a new typographic symbol to represent the new common European currency, the "Euro". Most existing character set standards and fonts did not include the symbol € representing the Euro. Character set standards were eventually changed and new fonts, conforming with the revised character set standards, were developed and distributed.

Another approach to the introduction of the new Euro symbol would have been to distribute a glyphlet representing the € symbol, including glyph attributes and character attributes. The glyphlet can be stored by an operating system. In one implementation, a text engine can build a menu or palette indicating characters available for use in a text document from a collection of one or more such glyphlets stored by the operating system. The text engine can query the glyphlets and generate corresponding menu items, which a user can use to select the indicated character. The character may be indicated by a graphic representation, *i.e.*, a glyph shape rendered based on glyph attribute information obtained from the corresponding glyphlet, or by one or more character attributes unique to the glyphlet, such as the character name. By selecting a menu item, a user can insert a reference to a glyphlet corresponding to the character indicated by the menu item into a text document.

In another implementation, a menu item can be associated with one or more character attributes, rather than associated with a glyphlet. If the menu item is selected, then based on the one or more associated character attributes, the text engine can generate a query and query a collection of one or more glyphlets, which can be stored by the operating system. If a match is found, that is, a glyphlet including one or more character attributes satisfying the query, then the glyphlet can be used to render a glyph shape or otherwise process the glyphlet.

To illustrate the above, consider the Euro example. Figure 4 shows a process for constructing a text document included a reference to a glyphlet. In this example, the word processing application has a menu of characters, which includes a representation of the Euro symbol, such as a graphic image of the symbol, the name of the symbol, *i.e.* Euro, or some other attribute clearly identifying the Euro (Step 405). User input is received selecting the Euro, for example, by highlighting and clicking on the menu item representing the Euro (Step 410).

A glyphlet corresponding to the selected character is then identified (Step 415). For example, the menu item can be associated with a pointer that can be followed to a memory location where the glyphlet is stored. Alternatively, the one or more target attributes can be stored at the memory location, from which a query can be formed. The query can be used to query a collection of glyphlets to identify a target glyphlet having one or more attributes satisfying the query.

Once a glyphlet is identified, a reference to the identified glyphlet is inserted into the text document (Step 420). As discussed above, the reference can be an out-of-band value or one or more in-band values. A glyph shape rendered from the glyph attributes included in the target glyphlet can be displayed in a display of the text document, for example, on a monitor or by a printer. The reference can be subsequently used by a text engine processing the text document to identify the target glyphlet.

A recipient of the text document processing the text document using a different text engine and a different computer may find a reference in the text document meaningless, particularly if the recipient's text engine does not have access to the glyphlet. To avoid this situation, the glyphlet can be embedded within the text document and thus accessible to a recipient of the text document. In another implementation, the reference can provide a text engine processing the document information about where to obtain a corresponding glyphlet, for example, an address where the glyphlet can be retrieved from a server.

As discussed above, if a character set standard is revised, then a font constructed under a previous version of the character set standard may be outdated and considered incomplete and non-conformant. To bring a font into conformance with the revised character set standard requires revisions to the font, and distribution and installation of a new font to users of the existing font. By using one or more glyphlets, the glyph shapes accessible by a font can be expanded without requiring a revised font to be distributed and installed by font users.

One approach to expanding the glyph shapes accessible by a font is to provide font users with a collection of one or more glyphlets corresponding to changes made to the underlying character set standard. For example, when a new character is added to the character set standard, rather than distribute a revised font that includes a glyph image for the new character and shares an encoding value mapping to the corresponding new character's attributes in the character set standard, a new glyphlet for the new character is distributed to font users that can be used in conjunction with the existing font. The new

glyphlet includes glyph attributes for rendering a glyph shape representing the new character, with any stylistic features necessary to conform to the font typeface. The new glyphlet also includes character attributes corresponding to the new character. The character attributes mirror those included in the revised character set standard

5 corresponding to the new character, but are accessible by querying the new glyphlet, rather than by an encoding value mapping to the character set standard. Thus, the font in conjunction with the new glyphlet performs just as if a revised font including a glyph image for the new character and a corresponding mapping to the new character in the character set standard were being used. In this manner, additional glyph shapes can

10 easily, quickly and inexpensively be accessible without necessitating issuance of a revised form of the font.

Another approach to expanding the glyph shapes accessible by a font using a glyphlet is to include a mapping to one or more references in a font. The one or more references can be used to identify one or more glyphlets. As discussed above, the

15 reference can be uniquely associated with a glyphlet, or can identify one or more target attributes that can be used to identify a target glyphlet.

By way of illustrative example, consider the introduction of the new typographic symbol to represent the "Euro", discussed above. Before the symbol € became representative of the Euro, it was known that a new common currency would be

20 established, and that a new symbol would likely be necessary to represent the currency. Accordingly, there was a time period during which font manufacturers released fonts knowing that they would soon be rendered out-dated by the introduction of the new currency symbol. A character set standard from which a font was constructed may have been revised to include an encoding value and character attributes for the Euro, although

25 the representation of the Euro symbol had not yet been determined. One solution would have been to include a mapping in the font from the Euro encoding value to a reference to a set of attributes, which attributes could later be used to form a query from which a Euro glyphlet could be identified. For example, the set of attributes could include the "currency symbol" attribute. Later, once the symbol had been determined, the font

30 manufacturer could provide a new glyphlet, including an appropriate set of glyph attributes and character attributes, for example, a name attribute having the value "Euro" and a "currency symbol" attribute.

FIG. 5 shows a process for expanding the glyph shapes accessible by a "dynamic" font. Using the above illustrative example, the first step includes receiving a dynamic

font, that is a font that includes a mapping to a reference that can be used to identify a glyphlet having a "currency symbol" attribute (Step 505). The font does not include an encoding value mapped to a glyph within the font for the € symbol, because the font was released before it was known that the € symbol would represent the Euro.

5       Once it is known the € symbol represents the Euro, the font manufacturer creates a glyphlet representing the Euro character, including a set of glyph attributes and character attributes. The Euro glyphlet is provided to users of the font and installed (e.g. stored in memory), rather than requiring installation of a new font altogether (Step 510). A user can then create a text document including the € symbol, for example, by choosing the  
10       symbol from a drop down menu on a text processing application as described above. The application inserts in the text document the reference to a set of one or more attributes, such as the "currency symbol" attribute, from which the Euro glyphlet can be identified.

      A text engine processing a text document including the € symbol encounters the reference and recognizes the reference as referring to a set of attributes from which a  
15       glyphlet can be identified. The text engine then generates a query based on the set of attributes (Step 515). A cache or collection of glyphlets is queried (Step 520). The text engine identifies the Euro glyphlet from the collection of glyphlets as having attributes satisfying (i.e., matching to some degree) the query (Step 525). The text engine can then render the glyph shape, in this example the € symbol, or otherwise process the glyphlet as  
20       if a glyph had been included in the font's encoding and mapped to the underlying character set standard.

      In one implementation, a font can be formed entirely of a mapping to references from which glyphlets can be identified, as described above in the context of the Euro symbol example. By using the font in conjunction with a collection of one or more  
25       glyphlets, glyphlets referenced by the font can be accessed to render glyph shapes or otherwise process the glyphlets based on the glyphlet's character and glyph attributes.

      The first time a text engine encounters a reference and identifies a glyphlet based on the reference, the text engine can store an identifier for the glyphlet associated with the reference. The next time the reference is encountered, text engine can identify the  
30       glyphlet based on the stored identifier, without having to perform another query. For example, if the glyphlet includes a name, the text engine can associate the name of the glyphlet with the reference. The next time the reference is encountered, the text engine can use the glyphlet's name to access the glyphlet without requiring a second query.



To summarize the above in the context of a font-rendering system, consider a font-rendering system tasked to render a desired glyph shape from a font. If the font includes a mapping to a glyph definition, the system displays a glyph image. If the font includes a mapping to a reference from which a glyphlet can be identified, the system identifies the glyphlet based on the reference and displays a glyph image. The reference can be associated with either a glyphlet or a set of attributes from which a glyphlet can be identified, or can include the glyphlet, as discussed above. If the reference identifies a set of attributes, the system can form a query based on the attributes, and search a store of glyphlets to find a glyphlet having attributes best matching the query. The system can task an external mechanism to perform the search. The external mechanism returns to the system a glyphlet that includes attributes satisfying the query. The system can cache the glyphlet in memory and display the glyph image, or otherwise process the glyphlet based on the glyph and/or character attributes.

In one implementation, a typeface can be formed entirely of glyphlets. A glyphlet-based typeface does not include any encoding values mapped to glyphs in a font and characters in a character set standard. Rather, the typeface is formed from a collection of one or more glyphlets that can be identified by a reference included in a text document, as described above. The reference can be directly associated with a glyphlet, or can be associated with one or more target attributes from which a glyphlet can be identified. Such a font can be useful for glyphs that are likely to change often.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose

microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; a magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

To provide for interaction with a user, the invention can be implemented on a computer system having a display device such as a monitor or LCD screen for displaying information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer system. The computer system can be programmed to provide a graphical user interface through which computer programs interact with users.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, the steps of the invention can be performed in a different order and still achieve desirable results.

What is claimed is:

1. A computer-implemented method for constructing a text document, the method comprising:
  - receiving user input selecting a character (410);
  - 5 identifying a glyphlet corresponding to the selected character (415), the glyphlet including a set of character attributes defining semantic information of the selected character and a set of glyph attributes defining appearance information for a glyph representative of the selected character; and
  - inserting a reference to the identified glyphlet into a text document (420).
- 10 2. The method of claim 1, wherein:
  - user input selecting a character (410) includes user input selecting a glyph shape representing the character.
3. The method of claim 1, wherein:
  - the reference to the identified glyphlet (420) includes one or more in-band
  - 15 values (110) defined in an encoding standard.
4. The method of claim 3, wherein:
  - the one or more in-band values (110) define one or more target attributes uniquely identifying the identified glyphlet in a collection of glyphlets.
5. The method of claim 3, wherein:
  - 20 the one or more in-band values (110) define the identified glyphlet.
6. The method of claim 1, wherein:
  - the reference to the identified glyphlet (420) includes one or more out-of-band values not defined in an encoding standard.
7. The method of claim 6, wherein:
  - 25 the one or more out-of-band values are associated with one or more target attributes uniquely identifying the identified glyphlet in a collection of glyphlets.
8. The method of claim 6, further comprising:
  - embedding the identified glyphlet in the text document;
  - wherein the one or more out-of-band values are directly associated with the
  - 30 identified glyphlet.
9. The method of claim 1, further comprising:
  - embedding the identified glyphlet in the text document.

10. The method of claim 1, wherein:

the set of character attributes includes one or more character attributes selected from the group consisting of character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character.

11. The method of claim 1, wherein:

the set of glyph attributes includes one or more glyph attributes selected from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline, and glyph kerning.

12. A glyphlet, comprising:

a data structure stored on a computer readable medium, the data structure including character data representing one or more character attributes defining semantic information of a character and glyph data representing one or more glyph attributes defining appearance information for a representation of the character.

13. The glyphlet of claim 12, wherein:

the one or more character attributes include one or more character attributes selected from the group consisting of character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character.

14. The glyphlet of claim 12, wherein:

the one or more glyph attributes includes one or more glyph attributes selected from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline, and glyph kerning.

15. A computer program product, tangibly stored on a machine-readable medium, comprising instructions operable to cause a programmable processor to:

obtain an electronic document including a string (100) that includes one or more references;

parse the string (100) to identify a reference;

based on the identified reference, identify a glyphlet including a set of character attributes defining semantic information of a character and a set of glyph attributes defining appearance information for a representation of the character; and

use one or more character attributes or glyph attributes for the identified glyphlet to process text in the electronic document.

16. The computer program product of claim 15, wherein:  
the string includes a plurality of references comprising one or more in-band values (110) defined in an encoding standard.
17. The computer program product of claim 16, wherein:  
5 instructions operable to parse the string to identify a reference include instructions operable to interpret a plurality of in-band values (110) to define the identified reference.
18. The computer program product of claim 17, wherein instructions operable to:  
interpret a plurality of in-band values (110) to define the identified reference include instructions operable to identify one or more target attributes; and  
10 identify a glyphlet based on the identified reference include instructions operable to identify a glyphlet in a collection of glyphlets based on the identified target attributes.
19. The computer program product of claim 17, wherein:  
the plurality of in-band values (110) define the identified glyphlet.
20. The computer program product of claim 17, wherein:  
15 the plurality of in-band values identify a location external to the electronic document from where the identified glyphlet can be retrieved.
21. The computer program product of claim 14, wherein:  
the string includes one or more references comprising one or more out-of-band values not defined in an encoding standard.
22. The computer program product of claim 21, wherein:  
the identified reference includes one or more of the out-of-band values.
23. The computer program product of claim 22, wherein:  
the identified glyphlet is embedded in the electronic document; and  
the one or more out-of-band values are directly associated with the identified  
25 glyphlet.
24. The computer program product of claim 22, wherein instructions operable to:  
identify a glyphlet based on the identified reference include instructions operable to identify one or more target attributes based on the identified reference and identify a glyphlet in a collection of glyphlets based on the identified target attributes.
25. The computer program product of claims 18 and 24, wherein:  
the collection of glyphlets is embedded within the electronic document.
26. The computer program product of claims 18 and 24, wherein:  
the collection of glyphlets is stored external to the electronic document.

27. The computer program product of claim 15, wherein:  
the set of character attributes includes one or more character attributes selected from the group consisting of character case, character category, character combining class, character directionality, character numeric value, mathematical character, character language, letter character, alphabetic character and ideographic character.
28. The computer program product of claim 15, wherein:  
the set of glyph attributes includes one or more glyph attributes selected from the group consisting of glyph shape, typographic weight, typographic width, slant, number of strokes, glyph metrics, typeface name, glyph baseline and glyph kerning.
29. The computer program product of claim 15, further comprising instructions operable to:  
retrieve the identified glyphlet from a memory external to the electronic document.
30. The computer program product of claim 29, wherein:  
the identified glyphlet is retrieved from a collection of glyphlets.
31. The computer program product of claim 15, further comprising instructions operable to:  
retrieve the identified glyphlet from storage embedded within the electronic document.
32. The computer program product of claim 31, wherein:  
the identified glyphlet is retrieved from a collection of glyphlets.

1/2

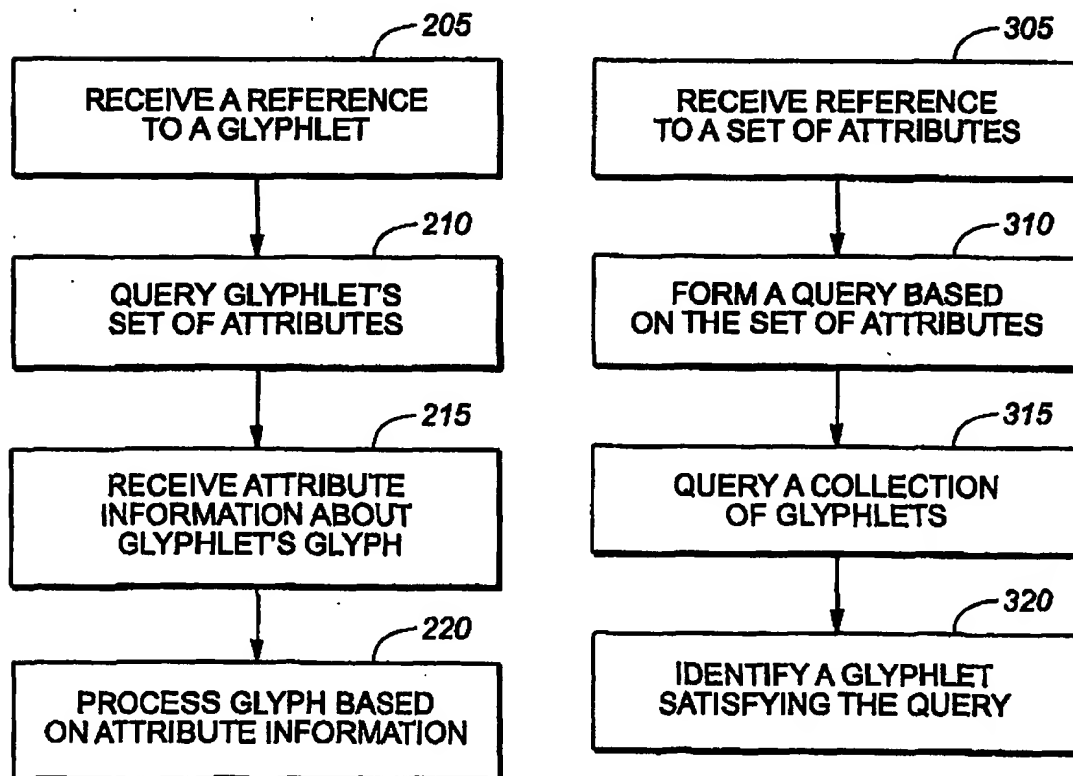
105                      110

The cost is <!attrib charcat="CurrencySymbol",  
 basedon="C", numStrokes=3, isnot="U+20A1",  
 isnot="U+20A2", baseDesign="TimesNewRoman",  
 weight="Regular", script="Latin",  
 languageGroup="WesternEuropean"> 1.99

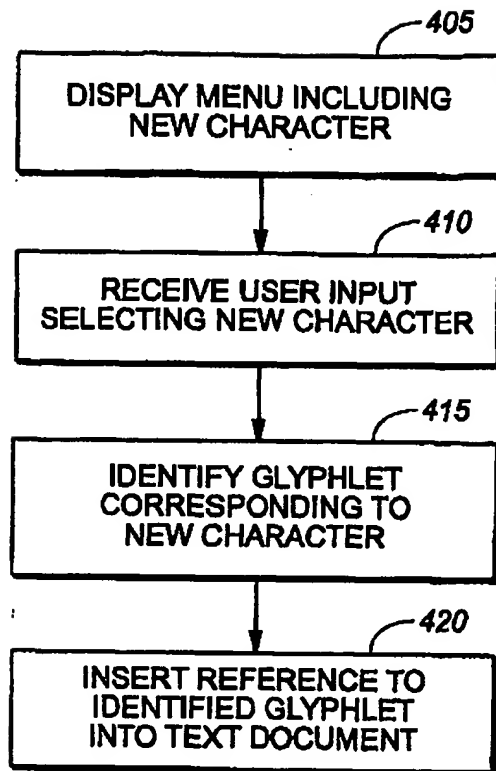
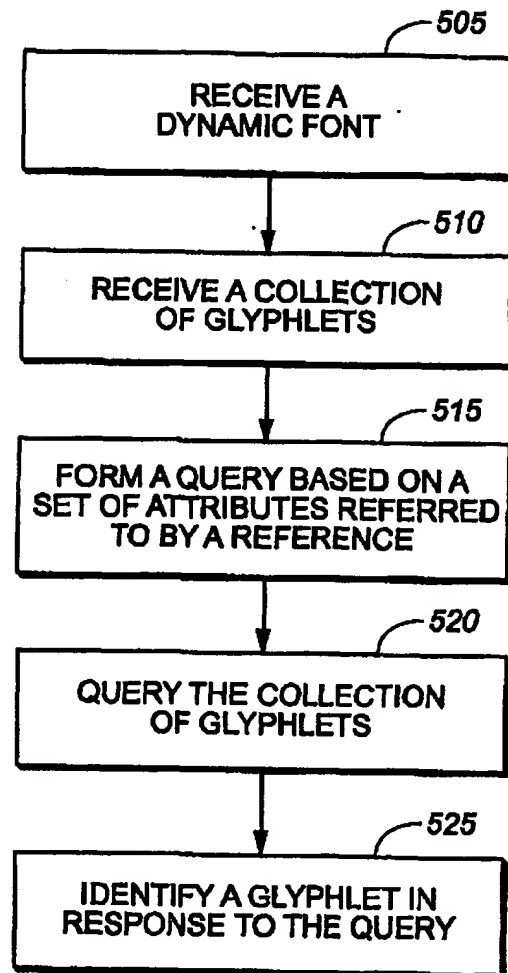
120

The cost is €1.99

100

**FIG. 1****FIG. 2****FIG. 3**

2 / 2

**FIG.\_4****FIG.\_5**



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**